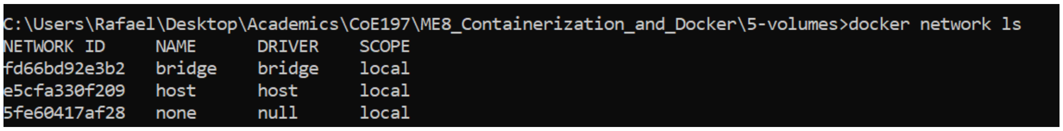
Maceda, Rafael
2016-01337
CoE 197S

Feature Activity 06

A. Listing Networks

Docker defines networks, which groups containers for interoperability and DNS functions.

To list networks, run `docker network ls`:

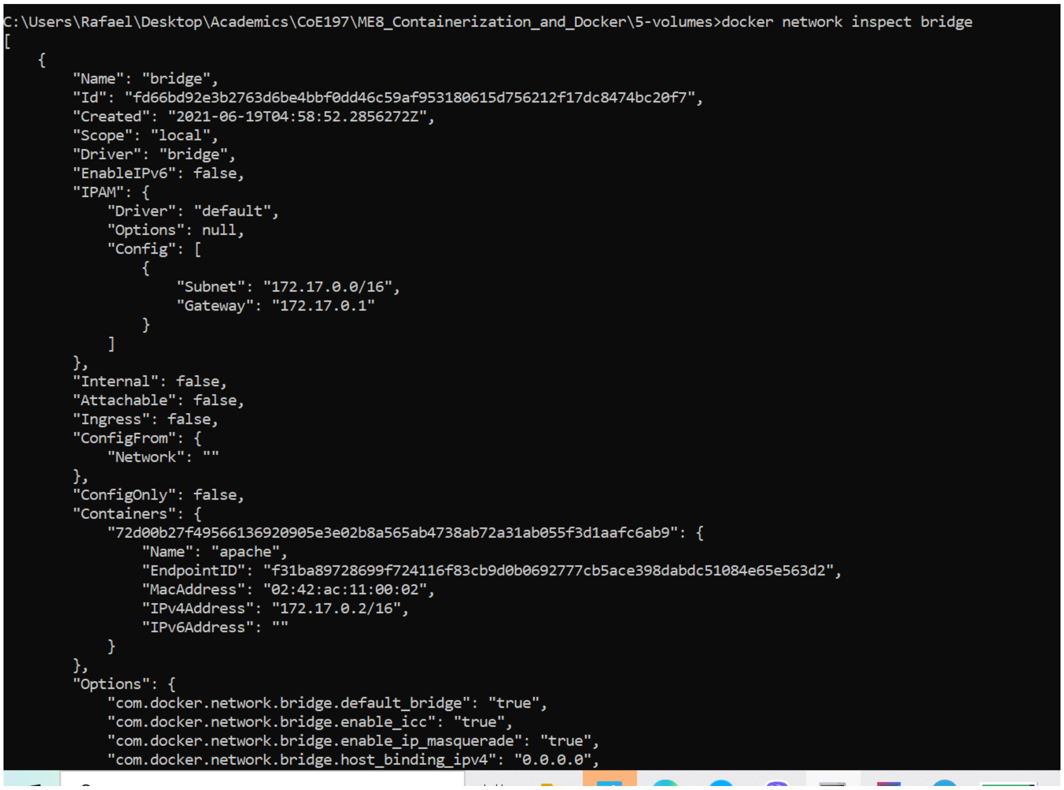


```
C:\Users\Rafael\Desktop\Academics\CoE197\ME8_Containerization_and_Docker\5-volumes>docker network ls
NETWORK ID     NAME      DRIVER    SCOPE
fd66bd92e3b2   bridge    bridge    local
e5cfa330f209   host      host      local
5fe60417af28   none      null      local
```

B. The default `bridge` network

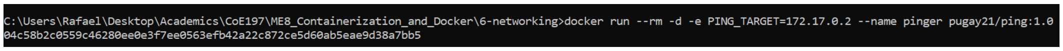
All new containers, if given no other configuration, will be automatically added the `bridge` network. This network acts as a pass through to your host's ethernet, so your Docker containers can access the internet.

We can inspect the `bridge` network by running `docker network inspect bridge`:



```
C:\Users\Rafael\Desktop\Academics\CoE197\ME8_Containerization_and_Docker\5-volumes>docker network inspect bridge
[
    {
        "Name": "bridge",
        "Id": "fd66bd92e3b2763d6be4bbf0dd46c59af953180615d756212f17dc8474bc20f7",
        "Created": "2021-06-19T04:58:52.2856272Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.17.0.0/16",
                    "Gateway": "172.17.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "72d00b27f49566136920905e3e02b8a565ab4738ab72a31ab055f3d1aafc6ab9": {
                "Name": "apache",
                "EndpointID": "f31ba89728699f724116f83cb9d0b0692777cb5ace398dabdc51084e65e563d2",
                "MacAddress": "02:42:ac:11:00:02",
                "IPv4Address": "172.17.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {
            "com.docker.network.bridge.default_bridge": "true",
            "com.docker.network.bridge.enable_icc": "true",
            "com.docker.network.bridge.enable_ip_masquerade": "true",
            "com.docker.network.bridge.host_binding_ipv4": "0.0.0.0",
```

You can see the container was added to the default network. Now let's add another `ping` container, and set it to ping our first.



```
C:\Users\Rafael\Desktop\Academics\CoE197\ME8_Containerization_and_Docker\6-networking>docker run --rm -d -e PING_TARGET=172.17.0.2 --name pinger pugay21/ping:1.0
04c58b2c0559c46280ee0e3f7ee0563efb42a22c872ce5d60ab5eae9d38a7bb5
```

Inspecting the logs for `pinger` we can see it was able to successfully ping the other container in the network. While IP address does work, it's very cumbersome and prone to error if addresses change. It would be better to use a hostname, specifically the container name `dummy`, to always resolve to the correct container.

Running `ping` with the `dummy` as the target: